المحاضرة السابعة

تصميم النظم

Systems Design

تصميم النظم
Systems Design

- **يمكن تعريف التصميم بانه** : عملية تمثيل للحل باستخدام مخططات وجداول وواجهة مستخدم وخوارزميات .
- **نشاطات مرحلة التصميم**

1. التصميم المعماري Architectural Design
2. تصميم واجهة المستخدم User Interface Design
3. تصميم البيانات Data Design
4. تصميم الاجراءات Procedural Design

التصميم المعماري
Architectural Design

- ويقصد به تجزئة المنظومة المراد إعدادها الى مكونات (**أجزاء برمجية**) في شكل هرمي ،
- والغرض هو تبسيط المنظومة المعقدة لأجل سهولة تصميم خوارزمياتها وتسهيل عملية توزيعها على المبرمجين،
- و من أهم الأدوات الشائعة والمستخدمة في التصميم المعماري هو **المخطط الهيكلي Structure Chart**

التصميم المعماري

المخطط الهيكلي (Structure Chart)

- **المخطط الهيكلي :** هو أداة تصميم رسومية توضح تركيبة (هيكلية) المنظومة في شكل هرمي.

❖ **ويبين هذا المخطط العناصر الآتية:**

- **الهيكلية الشجرية أو الهرمية** الكاملة للمنظومة كأجزاء برمجية (**Modules**).
- **العلاقة** بين هذه الأجزاء (**Relationship**).
- **البينيات** (**Interface**) وهي البيانات الممررة بين هذه الأجزاء البرمجية .
- **جزء التحكم** (**Control Module**) الموجود فى أعلى مستوي لقيادة الأجزاء البرمجية الأخري .

رموز المخطط الهيكلي

معناه	الرمز
جزء برمجي module	
بيانات data	
راية flag	
رابط connector	
قرار (أختيار) decision (op	
حلقة (تكرار) loop (iteration)	

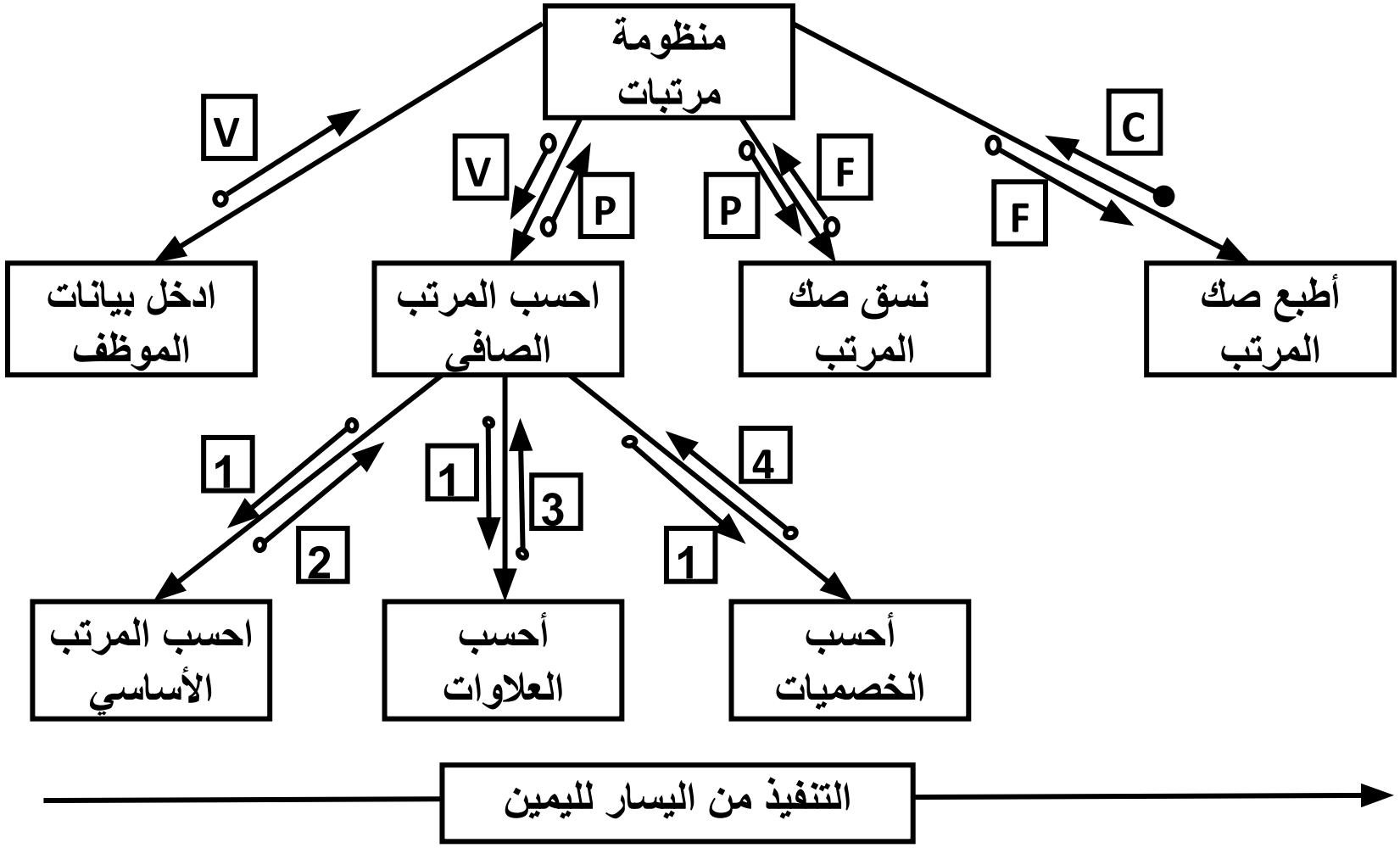
التصميم المعماري
(المخطط الهيكلي)

مثال 1 : المخطط الهيكلي التالي هو لمنظومة مرتبات ، ونلاحظ فى الشكل أن منظومة المرتبات تتكون من الأجزاء البرمجية التالية :

- **ادخل بيانات الموظف**
- **أحسب صافي المرتب**
- **نسق صك وقائمة المرتبات**
- **أطبع صك وقائمة المرتبات**

التصميم المعماري

المخطط الهيكلي (يتبع مثال ١)



F : صك وقائمة مرتبات منسقة	C : راية انتهاء العمل	P : معلومات لصك وقائمة المرتب	V : بيانات المرتب

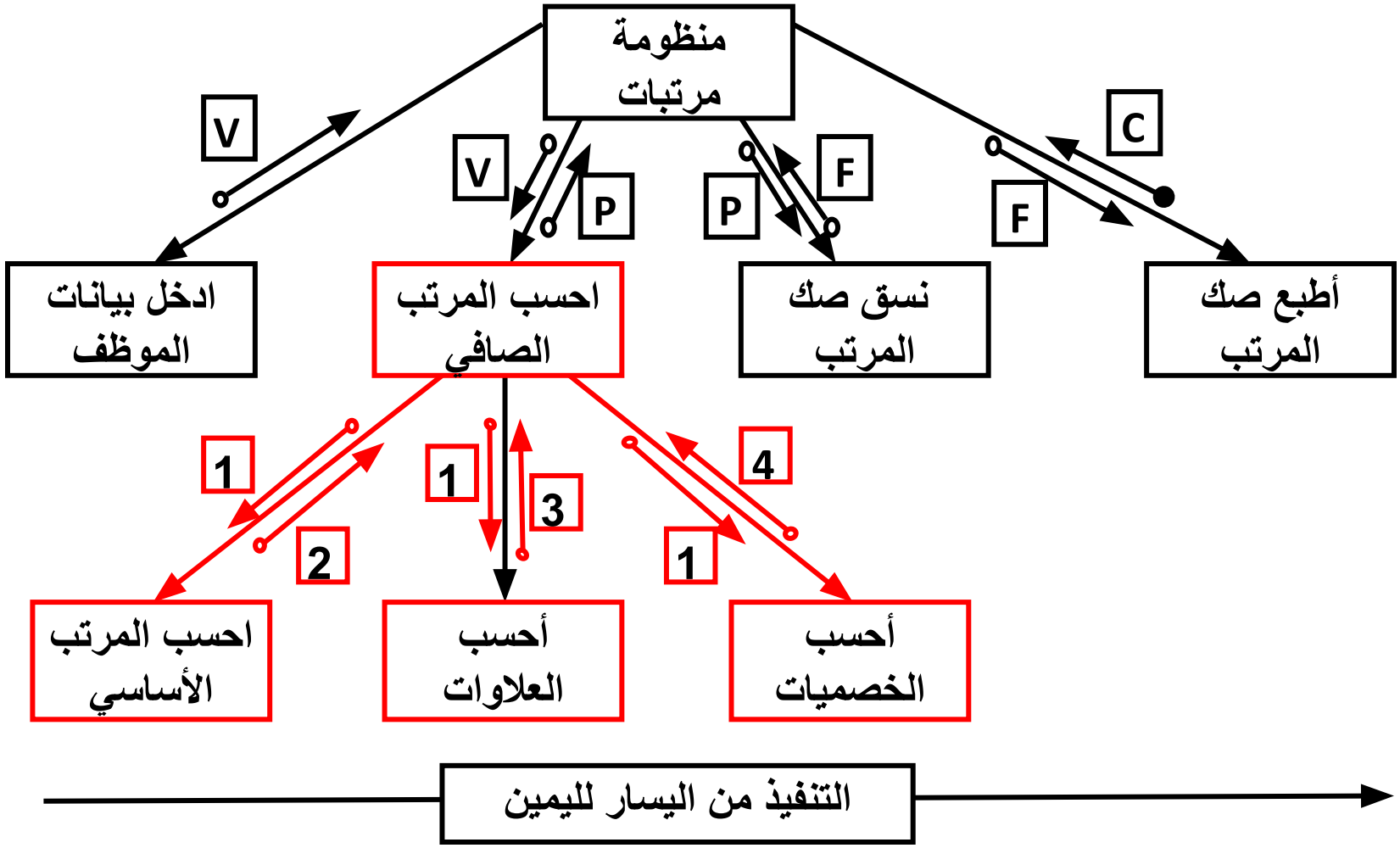
التصميم المعماري

المخطط الهيكلي (يتبع مثال ١)

- نلاحظ من المخطط السابق انه يمكن تقسيم الجزء (**أحسب المرتب الصافي**) الي أجزاء فرعية كما هو موضح فى الشكل،
- وبهذه الطريقة يمكن تقسيم باقي الأجزاء اذا ما أحتاج الأمر لتوضيح كل العمليات التي تتم على كل كل جزء .

التصميم المعماري

المخطط الهيكلي (يتبع مثال1)



4 : الخصميات	3 : العلاوات	2 : المرتب الأساسي	1 : بيانات المرتب

التصميم المعماري
(المخطط الهيكلي)

- **ملاحظة :** كما لاحظنا من المثال السابق (**مثال1**) ان العمليات يتم تنفيدها من اليسار لليمين بالترتيب وهذا النوع من المنظومات يسمي **منظومة تحولية** أى ان الادخال من اليسار ثم المعالجة فى المركز ومن ثم الاخراج فى جهة اليمين .

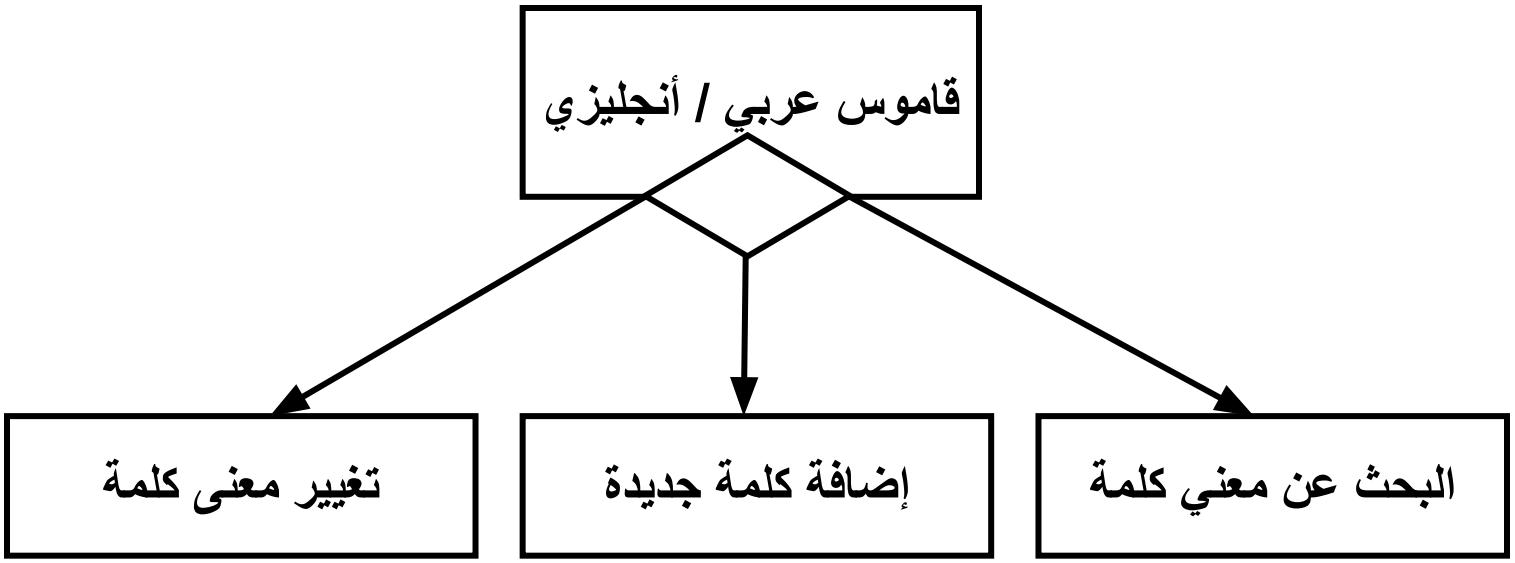
- أما فى **منظومة المعاملات** عادة ما يستعمل نوع واحد فقط من المعاملات فى الوقت الواحد، ولتوضيح هذا النوع من المنظومات ندرس المثال التالي.

التصميم المعماري

المخطط الهيكلي (مثال 2)

مثال2 : ارسم المخطط الهيكلي لمنظومة **قاموس عربي / انجليزى** لأداء الوظائف التالية :

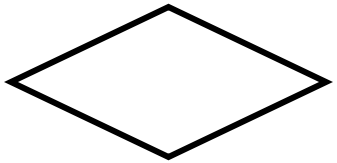
- البحث عن معني كلمة
- إضافة كلمة جديدة
- تغيير معني كلمة



التصميم المعماري

المخطط الهيكلي (يتبع مثال 2)

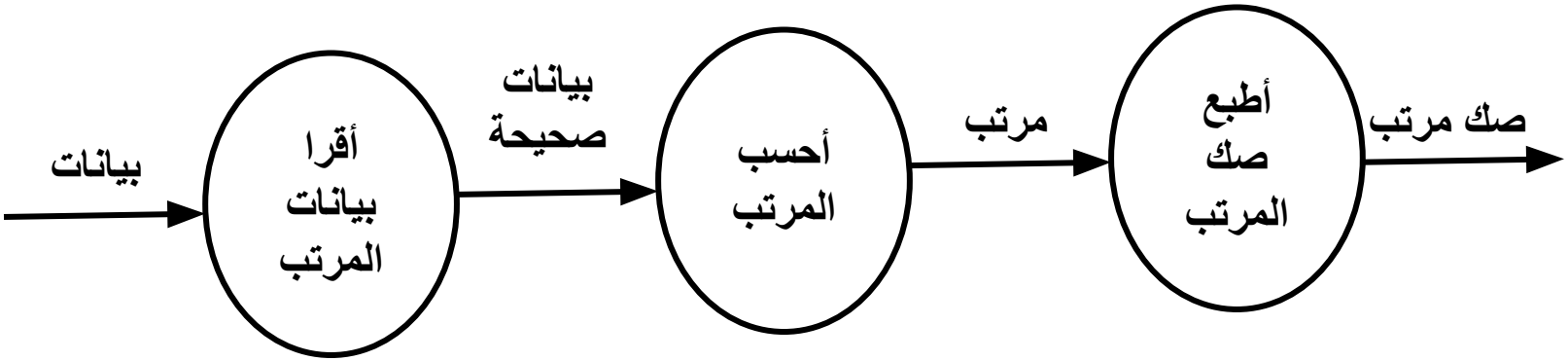
- نلاحظ من المخطط السابق ان الجزء البرمجي المتحكم يقوم بتحويل التحكم لإحدي عمليات المعاملة بناء على المعلومات الواردة الى هذا الجزء ،
- وذلك لأن المخطط المرتكزعلى المعاملة يستخدم دائما الرمز التالي :



- هذا الرمز هو أداة ضبط التحكم حيث تتفرع منه عدة معاملات ولكن يتم تنفيذ واحدة فقط اعتماداً على نوع المعاملة المطلوبة.

كيفية تحويل مخطط انسياب البيانات DFD الي مخطط هيكلي

مثال 3 : حول منظومة الرواتب المبينة بالشكل التالي من مخطط انسياب بيانات (DFD) الي مخطط هيكلي .



مخطط انسياب بيانات

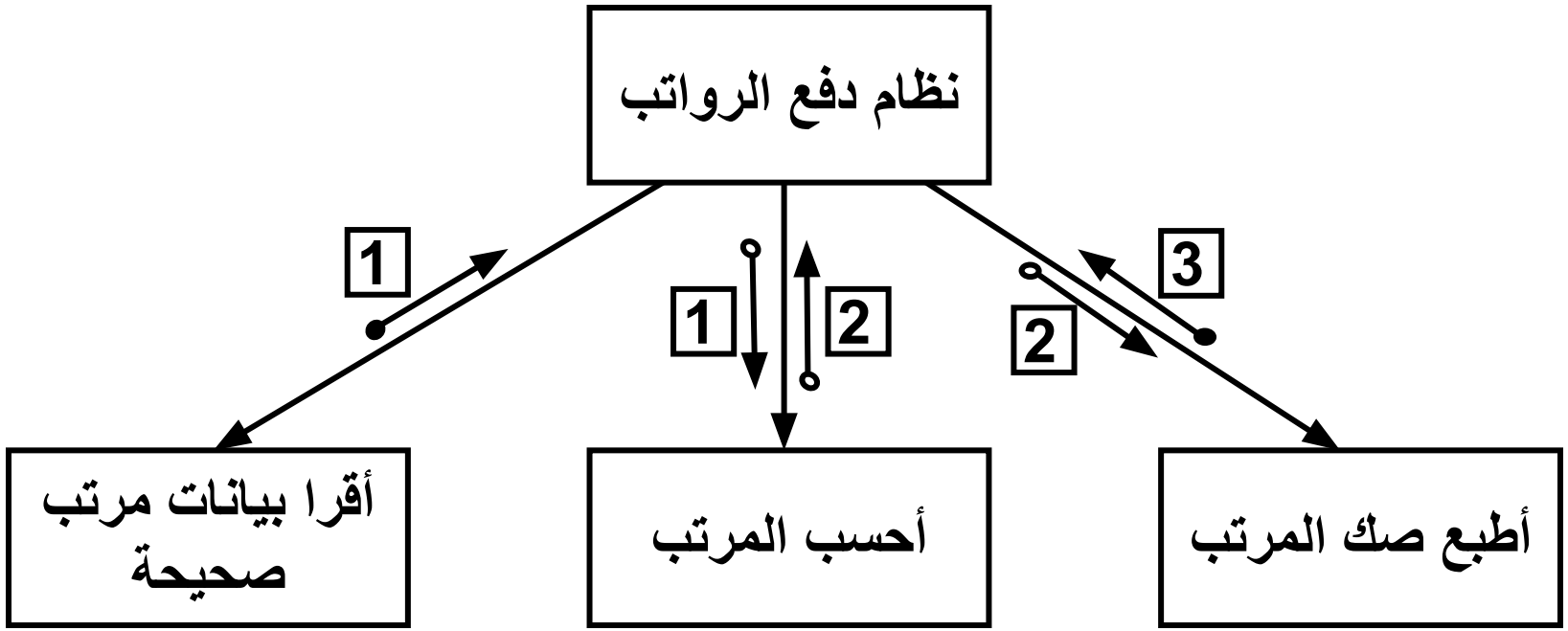
كيفية تحويل مخطط انسياب البيانات DFD الي مخطط هيكلي (يتبع مثال 3)

- **لتحويل مخطط انسياب البيانات الي مخطط هيكلي اتبع الخطوات التالية.**

1. كون **جزء تحكم (سواقة)** تحت أسم معين وليكن بأسم (**نظام دفع الرواتب**) في المخطط الهيكلي .
2. كل عملية فى DFD يتم تغييرها إلي **جزء برمجي (Module)** فى المخطط الهيكلي بنفس الاسم.
3. أربط هذه الأجزاء البرمجية بالسواقة مستخدماً **الوصلة (الأسهم).**
4. انسيابات البيانات في مخطط DFD يجب ان تتحول الي **رموز بيانات أو رايات** فى المخطط الهيكلي .

كيفية تحويل مخطط انسياب البيانات DFD الي مخطط هيكلي (يتبع مثال 3)

- النتيجة هذا الشكل:



3 : راية النهاية	2 : المرتب	1 : بيانات صحيحة

تصميم واجهة المستخدم
User Interface Design

- يتعلق هذا النوع من التصميم بالتعامل والتواصل بين المنظومة ومستخدميها ، ويتعلق تصميم الواجهة بتصميم المدخلات والمخرجات على الشاشة والتقارير على الورق.

- **يجب مرعاة عدة أمور عند تصميم الواجهة نسرد منها:**

1. جعل الواجهة سهلة التعلم والفهم .
2. جعل الواجهة تؤدي جميع الوظائف المنوطة بها .
3. أستخدام القوائم والرسومات (الأيقونات) قدر المستطاع.
4. الوضوح وعدم الالتباس فى المصطلحات .
5. توفير المساعدة Help.
6. جعل الواجهة تتعامل مع الأخطاء وتقلل حدوثها .
7. أستخدام مصطلحات بيئة عمل الزبون .

تصميم واجهة المستخدم
User Interface Design

- يمكن تصميم واجهة المستخدم بأستخدام أحد انواع التطبيقات البرمجية الحديثة مثل لغة البرمجة المرئية **Visual Basic** وغيرها من اللغات المرئية التي تتفاعل مع المستخدم عن طريق ما يعرف باسم واجهة المستخدم الرسومية (**GUI**) التي تتوفر فيها العديد من الميزات.

- ويمكن من خلالها تمثيل الكائنات رسوميا واستخدام صور الأزرار والنوافذ والقوائم الأفقية والعمودية التى تظهر على شاشة جهاز الحاسوب .

تصميم البيانات
data design

- الهدف الأساسي من تصميم البيانات هو تحويل البيانات التي تم إعدادها في مرحلة التحليل الي هياكل بيانات ليتم استخدامها في مرحلة البرمجة .

- **وعند تصميم البيانات يجب تجهيز الأتي :**
  - قاموس البيانات .
  - الكائنات والعلاقات بين هذه الكائنات .
  - الخصائص الخاصة بكل كائن .
  - مخطط الكائنات العلائقية ERD.

تصميم البيانات
(مخطط الكائنات العلائقية (ERD

- في مخطط الكائنات العلائقية يمكن أن يوجد كائنان مرتبطان بعلاقة مع خاصية ومفتاح رئيسي .

رموز مخطط الكائنات العلائقية	
كائن	مفتاح رئيسي
علاقة	خاصية

تصميم البيانات

(مخطط الكائنات العلائقية (ERD

مثال 1: يعتبر الزبون والكتاب كائنين. والعلاقة بينهما كما يلي :

- الزبون يستعير كتابا
- الزبون يستفسر عن كتابا
- الزبون يعيد كتابا

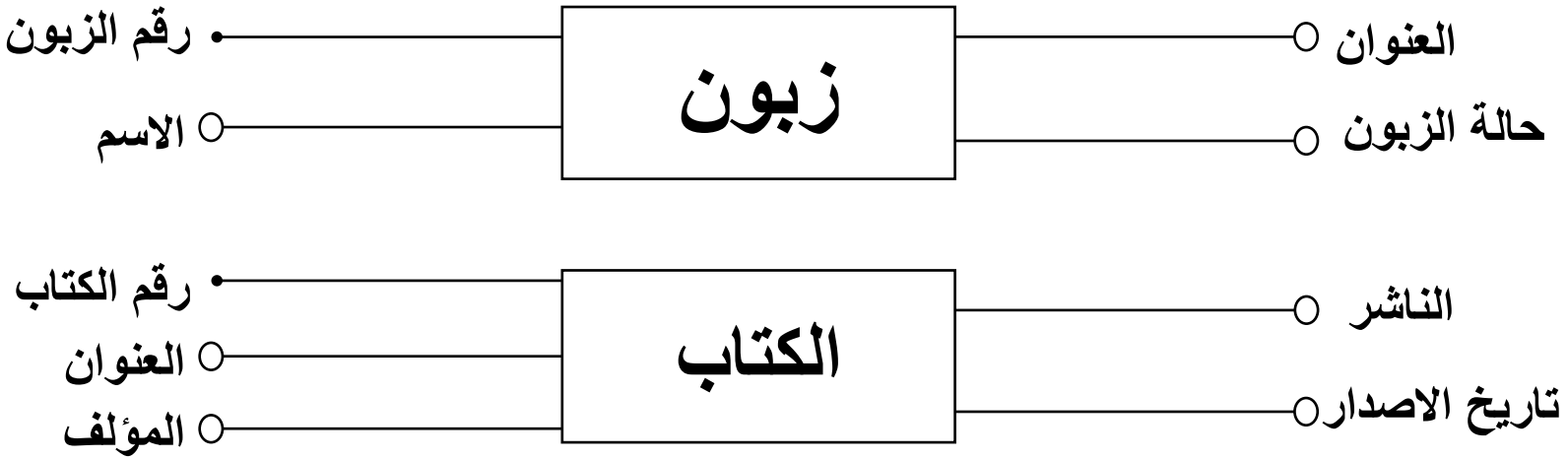
وعلى سبيل المثال يمكننا توضيح العلاقة كما يلي :

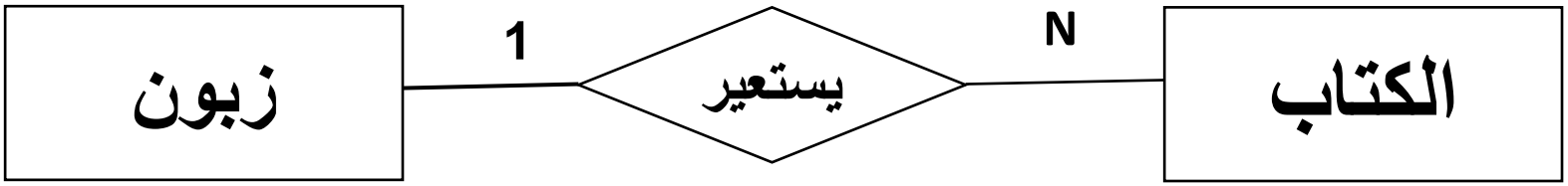
- الكائن زبون والخصائص التابعة لها .
- الكائن كتاب والخصائص التابعة له.
- والعلاقة بين الكائن زبون ، والكائن كتاب .

تصميم البيانات

مخطط الكائنات العلائقية ERD(يتبع مثال1)

- يمكن توضيح العلاقة بالمخطط كما يلي :





تصميم البيانات
(مخطط الكائنات العلائقية (ERD

- هذا المخطط يساعدنا في الحصول علي أساس لتكوين **جداول قاعدة البيانات** .
- حيث يمكن تكوين جدول لكل كائن
- ويمكن فى بعض الحالات تكوين جدول العلاقة

.

التصميم الاجرائي
procedural design

- **التصميم الاجرائي :** عادة ما يأتي بعد تجهيز التصميم المعماري وتصميم واجهة المستخدم .
- حيث يتم تجهيز مواصفات الإجراء لكل جزء برمجي تم ذكره فى التصميم المعماري بشئ من التفصيل .
- **وتستخدم أدوات لتمثيل هذه المواصفات والخوارزميات بأستخدام أحد الأدوات الاتية :**
  - **المخطط الانسيابي الهيكلى** Flow Chart Structured
  - **شبه الشفرة** Pseudocode

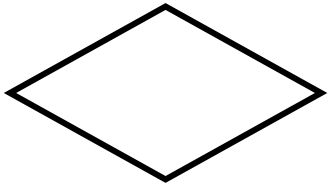
التصميم الاجرائي
المخطط الانسيابي الهيكلى Flow Chart

رموز المخطط الانسيابي الهيكلى

الرمز الطرفى Terminal

ادخال/ اخراج Input/Output

معالجة /عملية Process



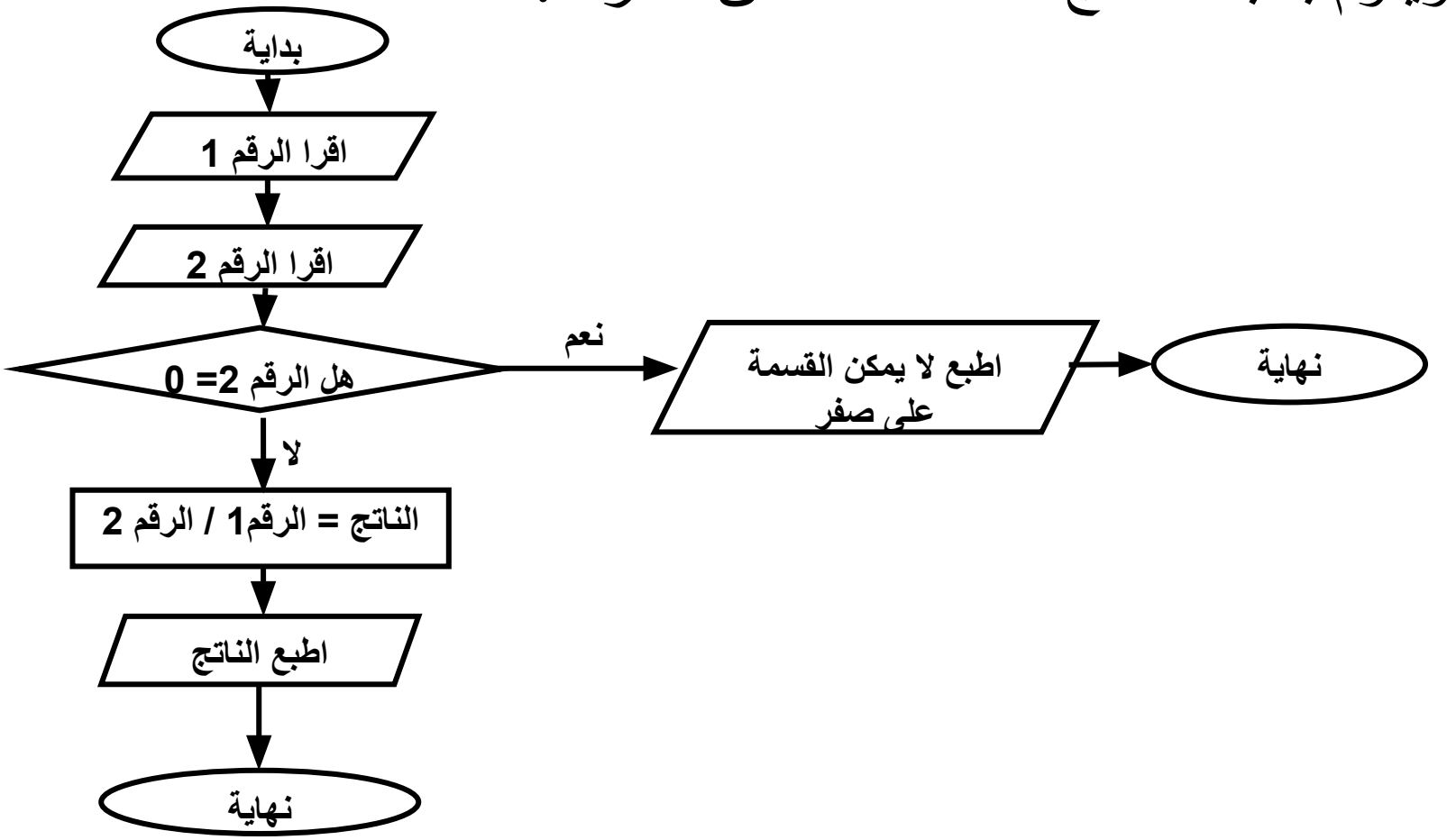
قرار/ اختيار Decision

خط اتجاه Flow Line

التصميم الاجرائي

المخطط الانسيابي الهيكلي Flow Chart

مثال: ارسم مخطط انسيابي هيكلي لبرنامج يقرا رقمين ، ومن ثم يقسمهم ، ويقوم بطباعة ناتج القسمة اذا تحقق الشرط.



التصميم الاجرائي

شبه الشفرة Pseudocode

- تعتبر شبه الشفرة أداة تصميم اجرائي نصي تبين منطق البرنامج (خطوات الخوارزمية) بلغة دقيقة ومفهومة مستخدمة تركيبات **البرمجة الهيكلية** الثلاثة:
  - التسلسلية.
  - والشرطية.
  - والتكرارية.

التصميم الاجرائي
شبه الشفرة Pseudocode

مثال: أكتب شبه الشفرة لبرنامج يحسب فاتورة الكهرباء (charge) باستخدام القراءة السابقة للعداد (last) والقراءة الحالية (pres) **حسب الشروط الآتية :**

- اذا كانت كمية الاستهلاك (**cons**) **اكبر** من 1000 كيلووات يكون سعر الكيلووات 50 درهم.
- اذا كانت كمية الاستهلاك (**cons**) **أقل أو يساوي** 1000 كيلووات يكون سعر الكيلووات 20 درهم.

التصميم الاجرائي
شبه الشفرة Pseudocode

الحل:

Read pres, last
Cons = pres – last

then <**If** cons 1000
Charge = (50*cons) / 1000
Else
Charge = (20*cons) / 1000
End **if**

Print pres, last, cons, charge
end

نقسم على 1000 بغرض التحويل من درهم الى دينار